

东南大学电工电子实验中心

实 验 报 告

课程名称： 数字电路实验

第 3 次实验

实验名称： 时序逻辑电路

院（系）： 电气工程学院 专 业： 电气工程及其自动化

姓 名： 王皓冬 学 号： 16022627

实 验 室： 401 实验组别：

同组人员： 实验时间：

评定成绩： 审阅教师：

一、实验目的

- 1、掌握时序逻辑电路的一般设计过程
- 2、掌握时序逻辑电路的时延分析方法，了解时序电路对时钟信号相关参数的基本要求
- 3、掌握时序逻辑电路的基本调试方法
- 4、熟练使用示波器和逻辑分析仪观察波形图

二、实验原理

1. 广告流水灯

- a) 用 D 触发器 7474 分别设计一个模 8 异步行波计数器和模 8 同步计数器，电路包含一个输出信号 F，当计数器计数值为“7”的时候， $F = 1$ ，其他计数值则 $F = 0$ 。在 Quartus 中进行时序仿真验证，并对两个仿真结果进行比较和分析
- b) 完成广告流水灯的设计，包含详细的设计过程和电路原理图
- c) 完成广告流水灯的硬件电路搭接

0. 输入信号与输出信号编码

输入信号

广告流水灯：用一个脉冲信号 P 作为输入（时钟），P 由低电平跃迁至高电平时（即从“0”变为“1”再变为“0”）产生一个脉冲，从而产生输入信号，使暗灯从左向右顺次移动。

输出信号

广告流水灯：用 B0 到 B7 作为第 1~7 个输出，分别对应第 1~7 个广告灯，输出为“1”时灯亮，为“0”时灯灭；F 代表完成一次循环，输出为“1”代表循环完成，为“0”时代表循环未完成。

中间值

广告流水灯：用 $Q_2 Q_1 Q_0$ 作为模八计数器的输出以及三八译码器的输入， $(Q_2 Q_1 Q_0)_2$ 组成三八译码器的三位二进制数地址。

1. 列出真值表

由于该广告流水灯为时序逻辑电路，故分为两个真值表：第一个真值表以输入信号 F 作为逻辑变量，列出中间变量的逻辑值变化（状态转换表）；第二个真值表以 $(Q_2 Q_1 Q_0)_2$ 输入，列出输出端的对应取值（输出方程）。（未列出取值的单元格取值为 0）

表 1 异步计数器真值表

输入	输出		
P	Q2	Q1	Q0
0	/	/	/
↑	Q_2'	Q_1'	Q_0'

表 2 同步计数器真值表

输入	输出		
P	Q2	Q1	Q0

0	0	0	0
↑	Q1	Q0	1

表 3 输出真值表

输入			输出								
Q2	Q1	Q0	B7	B6	B5	B4	B3	B2	B1	B0	F
0	0	0								1	
0	0	1							1		
0	1	0						1			
0	1	1					1				
1	0	0				1					
1	0	1			1						
1	1	0		1							
1	1	1	1								1

2. 逻辑化简

(1) 异步行波计数器

由真值表可得：

$$\begin{cases} Q_0^* = \overline{Q_0} \\ Q_1^* = \overline{Q_1} \\ Q_2^* = \overline{Q_2} \end{cases} \quad (1)$$

$$\begin{cases} CLK_0 = P \\ CLK_1 = \overline{Q_0} \\ CLK_2 = \overline{Q_1} \end{cases} \quad (2)$$

$$F = Q_2 Q_1 Q_0 = \overline{B_7} \quad (3)$$

其中，方程组（1）为状态方程，方程组（2）为驱动方程，方程（3）为输出方程。CLK_i为第 i 个 D 触发器的时钟变量。

(2) 同步计数器

由真值表可得：

$$\begin{cases} Q_0^* = \overline{Q_0} \\ Q_1^* = \overline{Q_1} \\ Q_2^* = \overline{Q_2} \end{cases} \quad (4)$$

$$\begin{cases} CLK_0 = P \\ CLK_1 = P \\ CLK_2 = P \end{cases} \quad (5)$$

$$F = Q_2 Q_1 Q_0 = \overline{B_7} \quad (6)$$

其中，方程组（4）为状态方程，方程组（5）为驱动方程，方程（6）为输出方程。CLK_i为第 i 个 D 触发器的时钟变量。

(3) 输出端

由真值表可得：

$$Bi = mi$$

其中 mi 是 $(Q_2 Q_1 Q_0)_2$ 组成的最小项。

3. 逻辑电路图

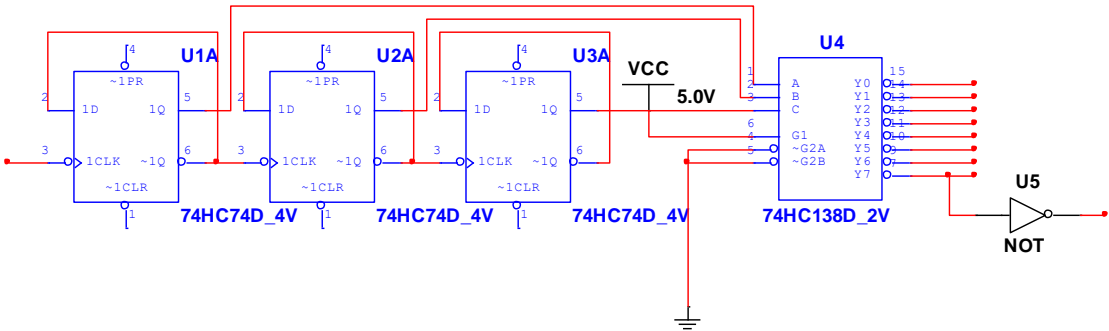


图 1 异步行波计数器

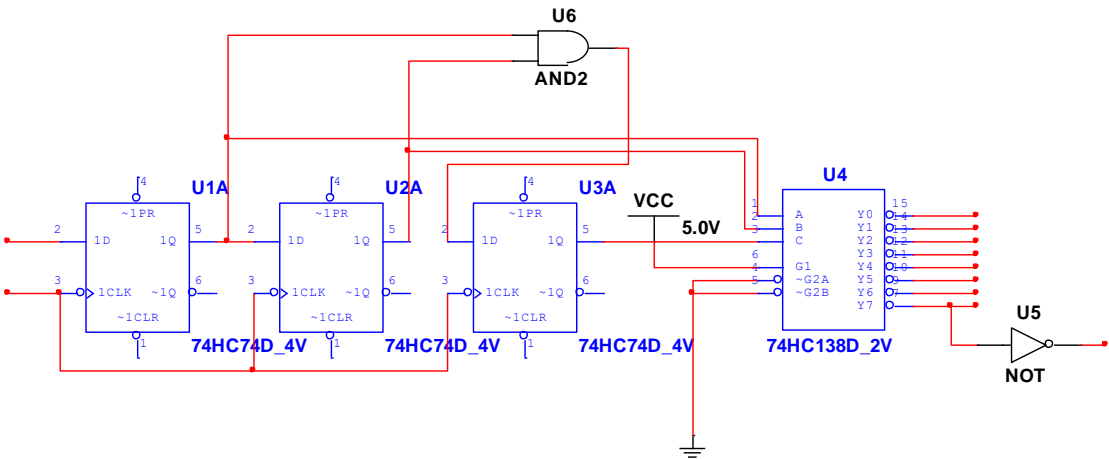


图 2 同步计数器

4. 硬件连接示意图

根据逻辑电路图所得硬件连接示意图如下：

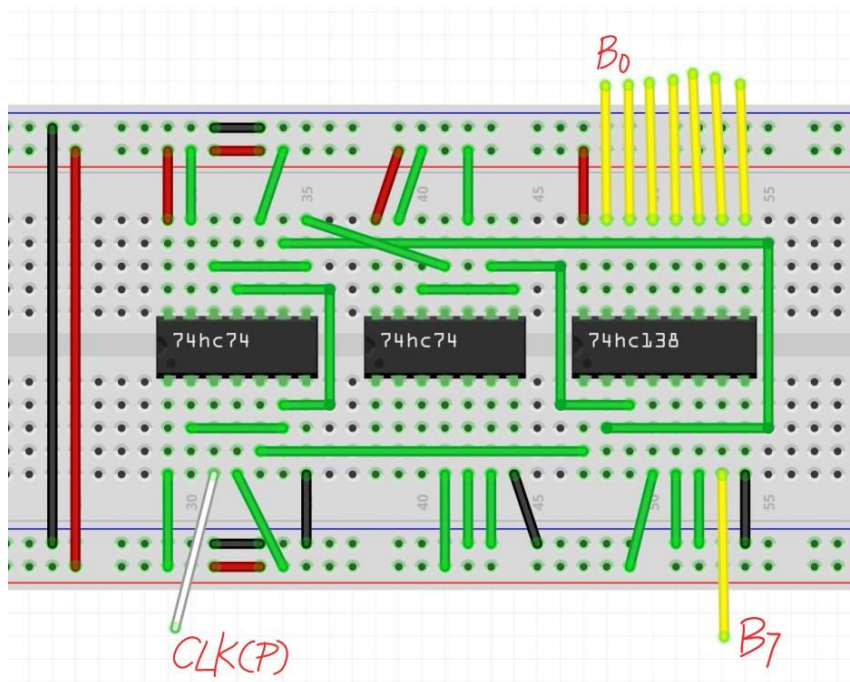


图 3 异步

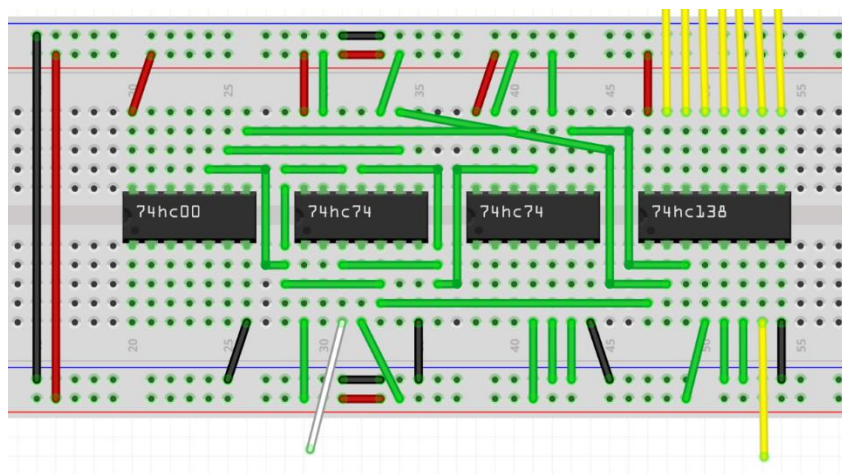
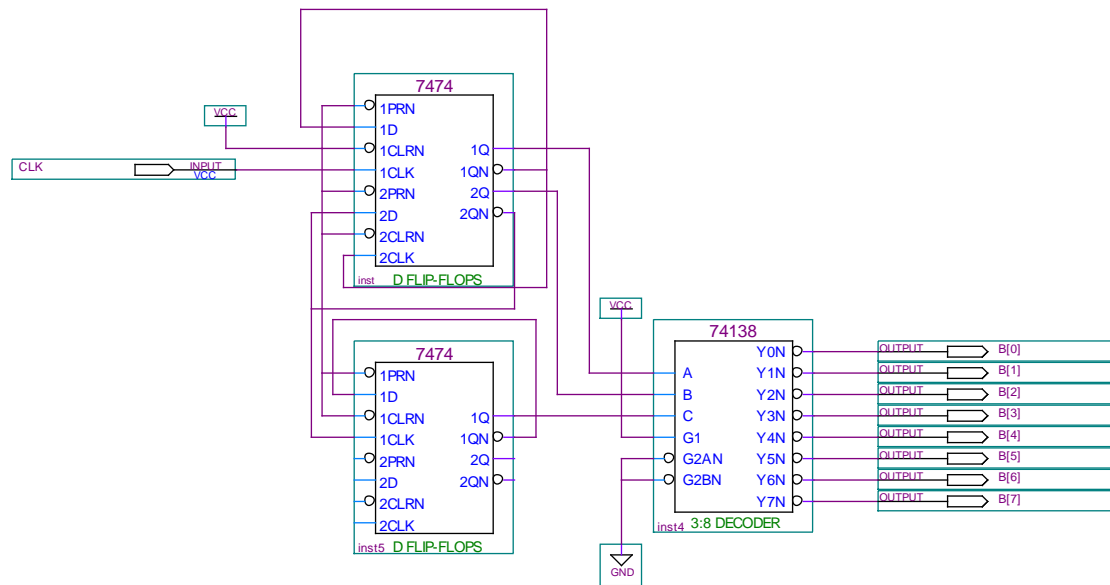


图 4 同步

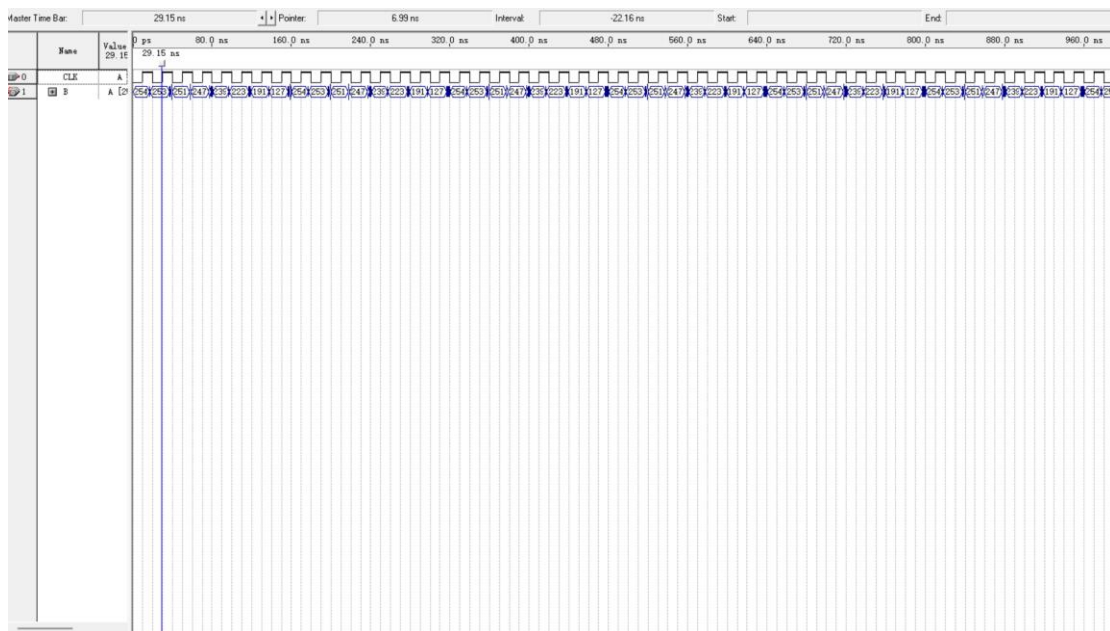
由于异步方案较为简单，实际搭建时采用异步电路。

5. 时序仿真

异步电路：

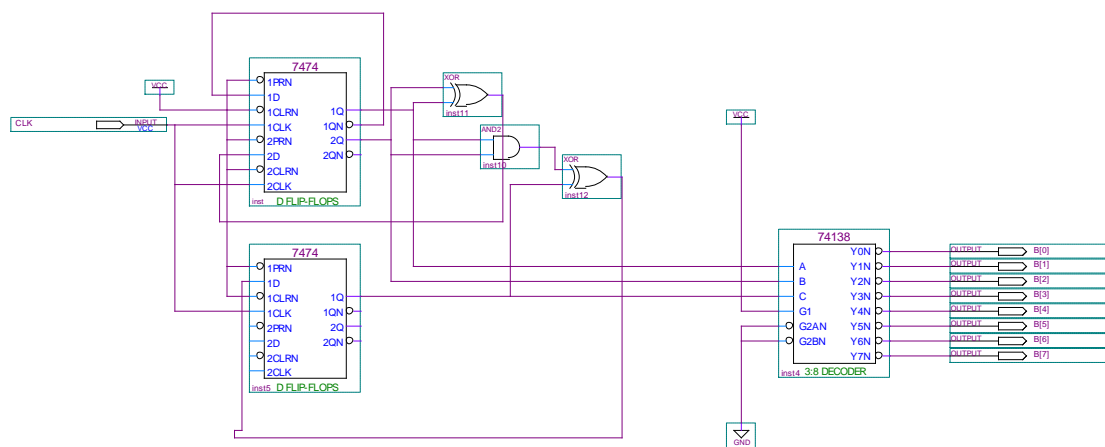


异步仿真结果：

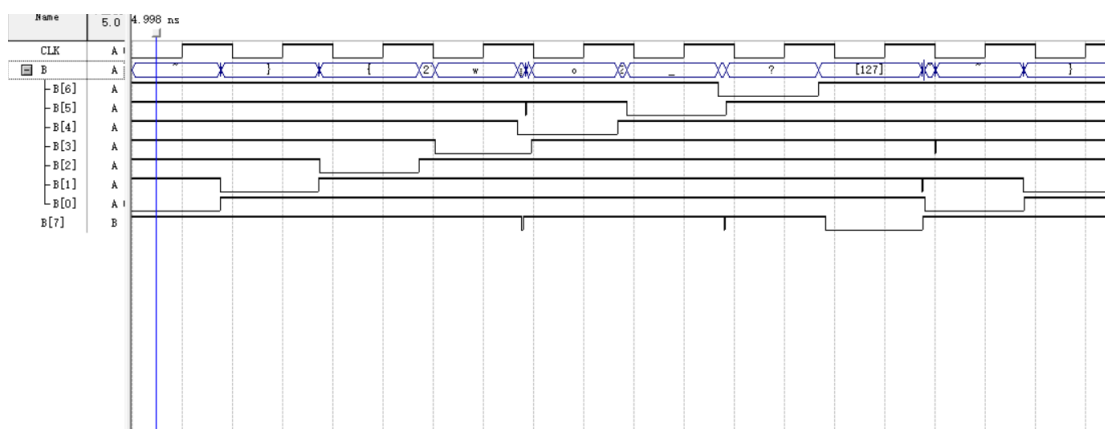


可以看到，初始为 8 位高电平（255），每次上升沿后约 7ns 延迟输出端波形产生变化，符合功能。

同步电路：



同步延迟:



可以看到, B[0]到 B[7]依次变为低电平, 每次上升沿后约 7.993ns 延迟输出端波形产生变化, 符合功能。相比而言。同步电路的毛刺更少。

6. 实验记录

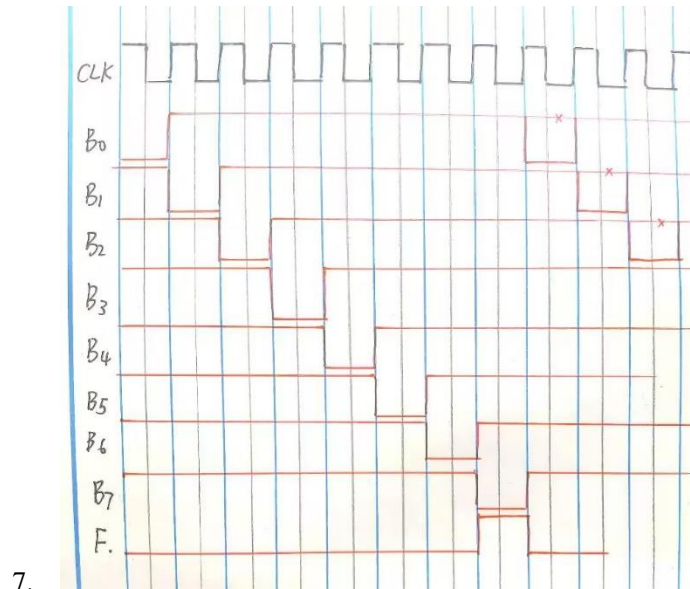
实验步骤: 首先利用脉冲按钮进行静态验证。按下按钮后, 灭灯从左向右依次移动, 最后一个灯灭时, 代表完成输出的灯亮, 随后重复上述现象。静态验证结果符合功能。

记录如下。

脉冲次数		输出								
/		B7	B6	B5	B4	B3	B2	B1	B0	F
	0								0	0
	1							0		0
	2						0			0
	3					0				0
	4				0					0
	5			0						0
	6		0							0
	7	0								1

其中为填入的逻辑项均为 1, 代表灯亮。

再利用双踪示波器进行动态验证。利用信号源产生频率 1kHz、高电平 5V、低电平 0V 的方波作为 CLK 信号, 依次显示(CLK,B0)、(B0,B1), ..., 波形, 记录结果。



波形说明，连续 CLK 信号输入时，B0~B7 依次跃变为低电平，B7 跃变的同时 F 跃变为高电平。动态验证结果符合功能。

该电路所有状态均为有效状态，*无需考虑自启动。*

遇到的问题及解决方法：无。

2. 序列发生器（第 11 周）

a) 分别用集成计数器 74161 的同步置“0”和异步清“0”功能实现模 10 计数器，在 Quartus 中进行时序仿真实验，并分析比较两种方法的区别

0 输入信号与输出信号编码

输入信号

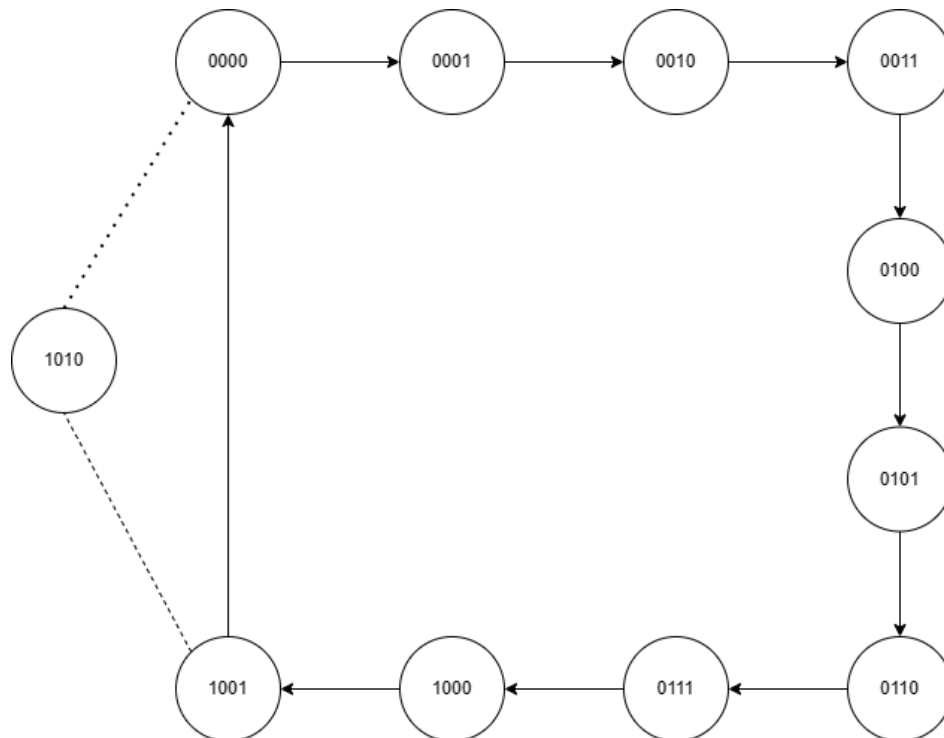
序列发生器：用 CLK 作为时钟信号。

输出信号

序列发生器：用 Y 作为输出信号。高电平代表信号“1”，低电平代表信号“0”。

1. 状态机

分析可知，同步置“0”功能应在“9”时置“0”，异步清“0”功能应在“10”时置“0”。对应状态机如下。



图中，实线为同步置 0 途径，虚线为异步置 0 途径。

2. 仿真电路图

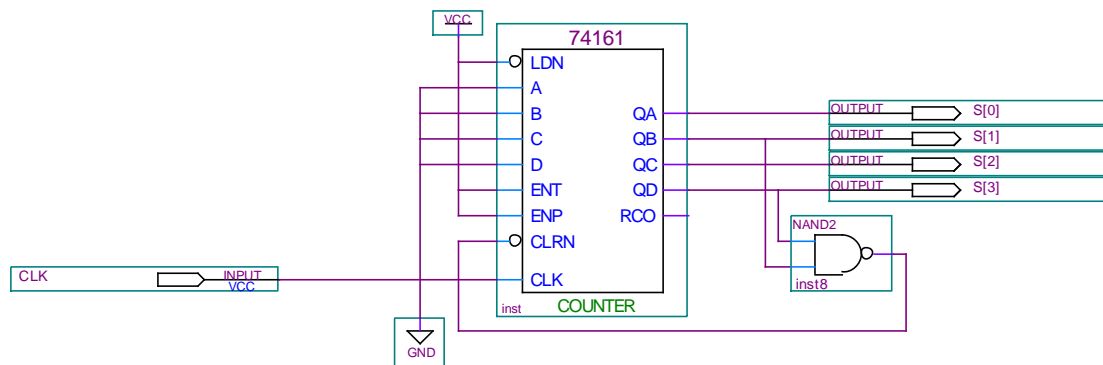


图 2 异步清 0 实现

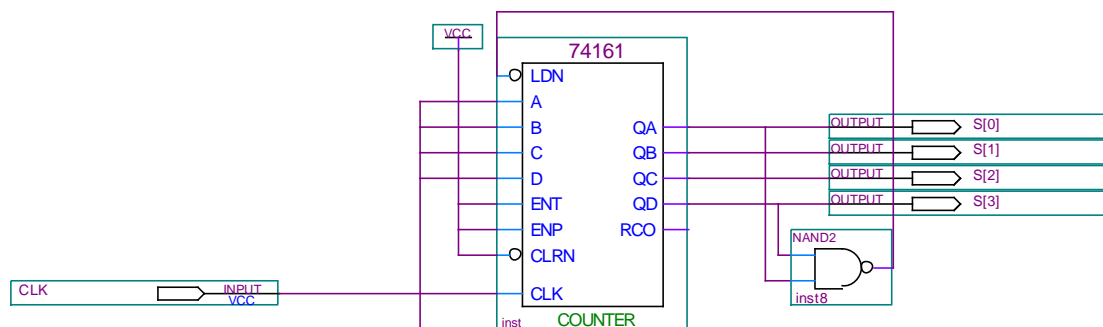
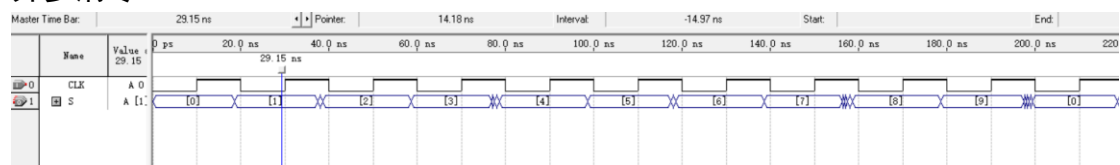


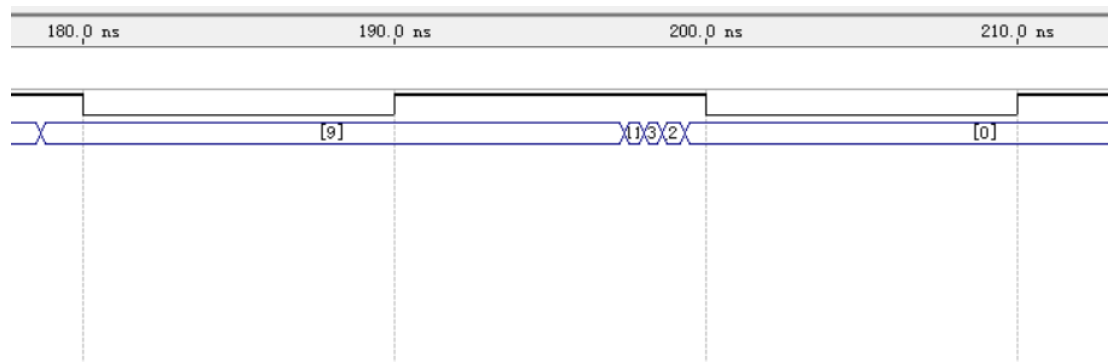
图 3 同步置 0 实现

3. 时序仿真

异步清零：

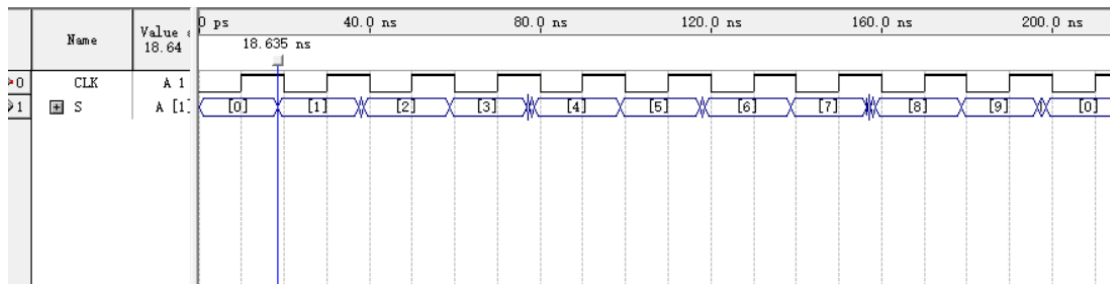


可以观察到，时钟上升沿的约 8.6ns 后，计数器作出反应。计数器由 0 计数到 9，对应 10 个数，符合功能。

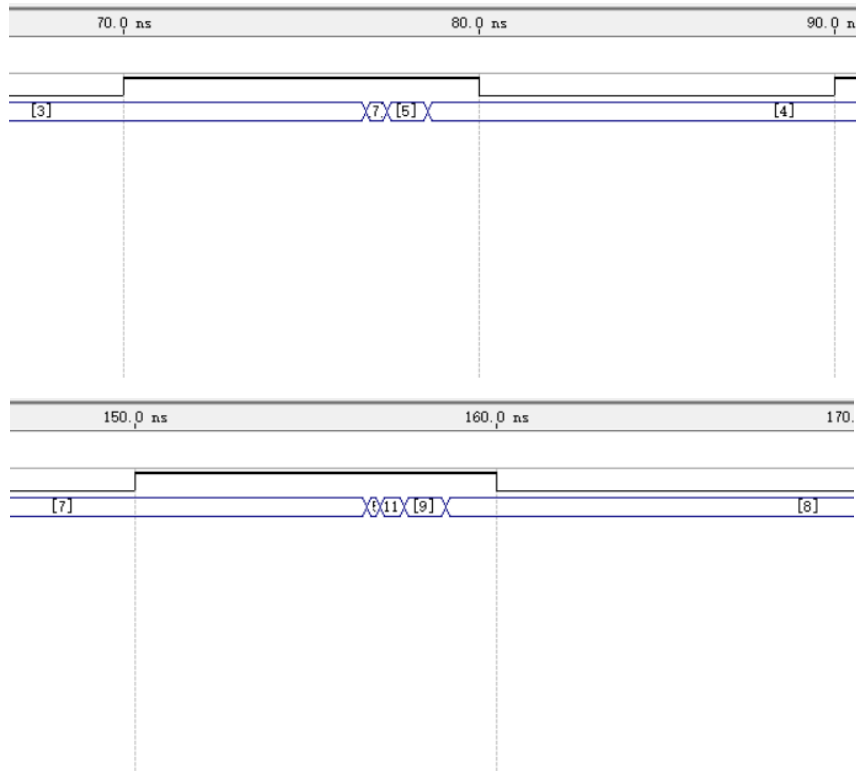


在 7->8、9->0 时，毛刺较多。

同步置零：



可以观察到，时钟上升沿的同样约 8.6ns 后，计数器作出反应。计数器由 0 计数到 9，对应 10 个数，符合功能。



同步置 0 的毛刺主要集中在 7->8 的跃变过程中。

仿真结果表明，异步清零与同步置零在延迟与毛刺产生方面均无明显差异。

b) 分别用集成计数器 74161 和 4 位双向移位寄存器 74194 实现图 3.1 所示的环形计数器，电路必须能自启动，并在 Quartus 中进行时序仿真验证

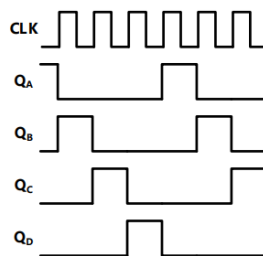


图 3.1 环形计数器波形

c) 完成两种方法实现序列发生器的设计方案，包含详细的设计过程和电路原理图

d) 完成两种方案序列发生器的硬件电路搭接。

1. 列出状态方程

$$Q_n = Q_{n-1}$$

$$Q_{n-1} = 0$$

其中， Q_n 是第 n 个输出信号（如 Q_B ）， Q_{n-1} 是第 $n-1$ 个输出信号（如 Q_A ）。

2. 逻辑化简

(4) 74161

分析可知，该计数器是一个模 4 计数器，只需将“0”“1”“2”“3”对应的二进制码接入对应输出即可。本实验采用同步置 0 的方式设计逻辑电路，即在计数器输出“3”的后一个 CLK 上升沿将预置数“0”同步置进计数器中。

$$\begin{cases} Q_A = \bar{S}_3 \bar{S}_2 \bar{S}_1 \bar{S}_0 \\ Q_B = \bar{S}_3 \bar{S}_2 \bar{S}_1 S_0 \\ Q_C = \bar{S}_3 \bar{S}_2 S_1 \bar{S}_0 \\ Q_D = \bar{S}_3 \bar{S}_2 S_1 S_0 \end{cases} \quad (1)$$

需要注意的是，为防止在“2”（0010）时 Q_A 输出“1”，应当完整写下 $S_1 S_0$ 的函数关系，而不能简单地用 \bar{S}_0 表示。 Q_B 、 Q_C 同理。由于实物搭接器件原因，将 $\bar{S}_3 \bar{S}_2$ 视为无效，即输出“5”（0101）与输出“1”（0001）等效。得到逻辑函数：

$$\begin{cases} Q_A = \bar{S}_1 \bar{S}_0 \\ Q_B = \bar{S}_1 S_0 \\ Q_C = S_1 \bar{S}_0 \\ Q_D = S_1 S_0 \end{cases} \quad (2)$$

这里用 74138 实现该函数组，74138 的最高位接“0”，实现数 4~7 的封锁。

(5) 74194

分析可知，设初态为 1000（ $Q_A=1$ ），以右移的方式设计电路，只需要在每次 CLK 上升沿到来时右移一位即可。为达成循环效果，只需将移位寄存器的输出反馈到它的串行输入端构成环形计数器即可。

3. 逻辑电路图

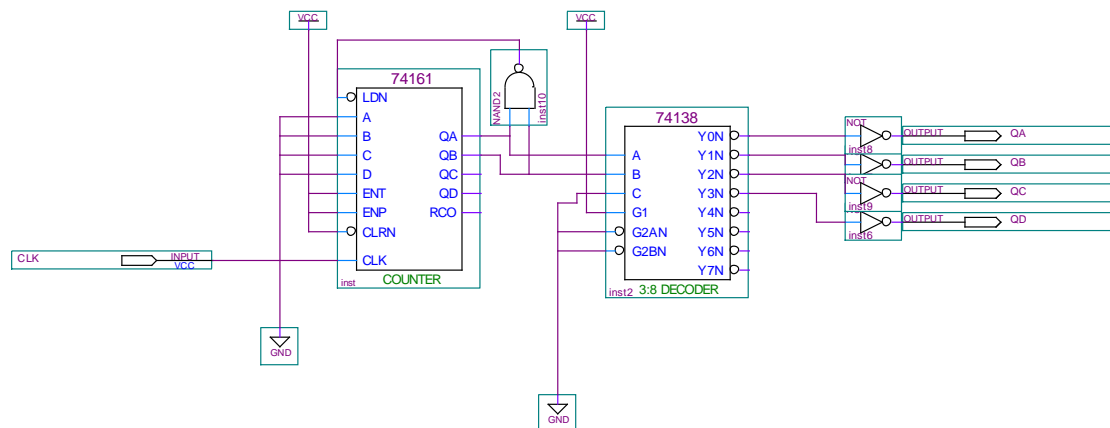


图 4 74161 实现

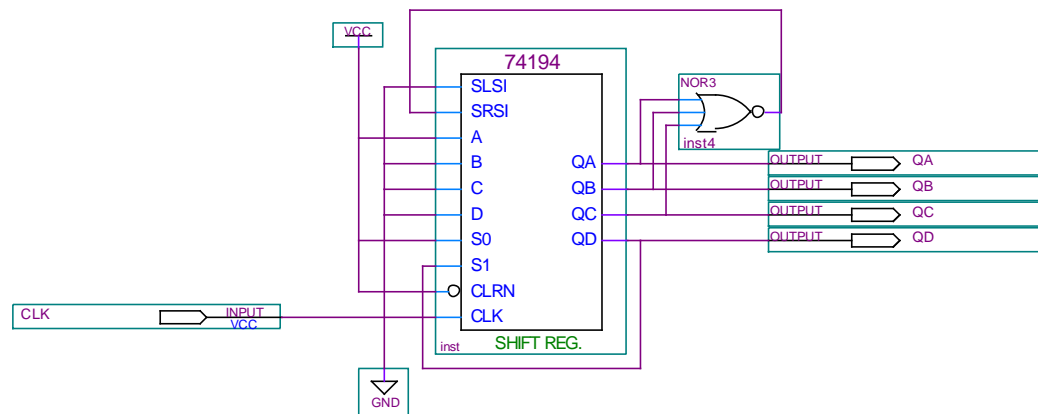
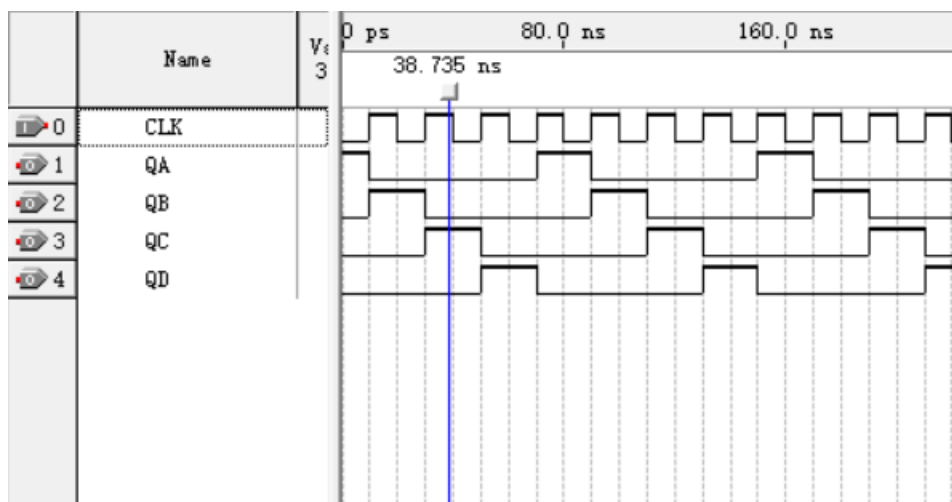


图 2 74194 实现

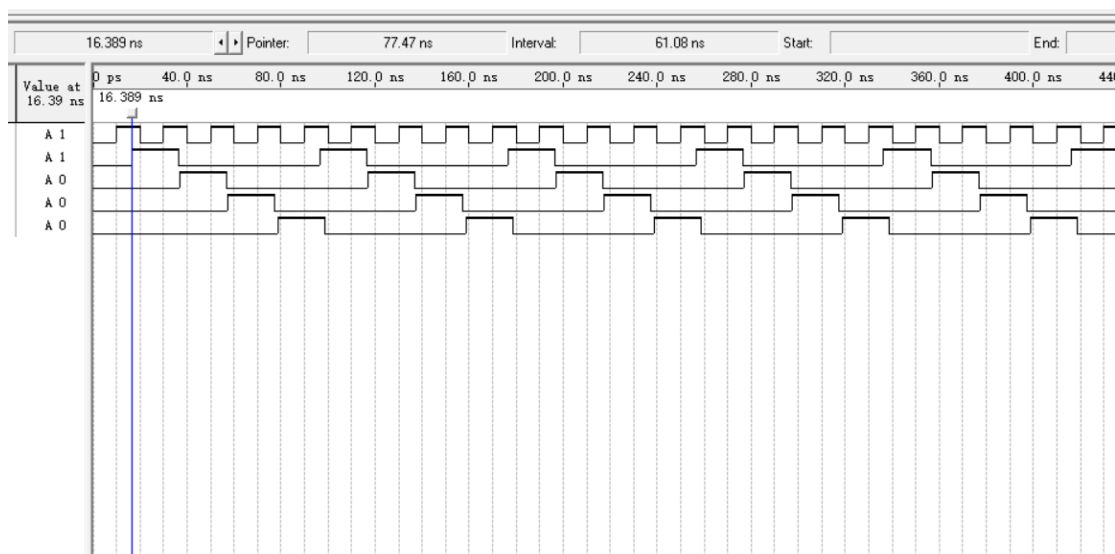
4. 时序仿真

74161:



观察到，电路能正确循环。

74194:



可以看到，输出正确，延迟约为 6.389ns。

e) 分别用 MSI 计数器和移位寄存器设计一个具有自启动功能的 101001 序列信号发生器

- (1) 写出设计过程，画出电路逻辑图
- (2) 搭接电路，并用单脉冲静态验证实验结果
- (3) 加入 TTL 连续脉冲，用示波器观察观察并记录时钟脉冲 CLK、序列输出端的波形。

1. 列出真值表

74161:

Q_2^n	Q_1^n	Q_0^n	Q_2^{n+1}	Q_1^{n+1}	Q_0^{n+1}	Y
0	0	0	0	0	1	1
0	0	1	0	1	0	0
0	1	0	0	1	1	1
0	1	1	1	0	0	0
1	0	0	1	0	1	0
1	0	1	0	0	0	1

其中，自启动功能是通过计数循环达到“000”实现的，例如“9”（1001）会逐个计数到“15”（1111），再返回“0”（0000）。

74194:

有效 状态	顺序	Q_C	Q_B	Q_A	SR	S_1	功能
	0	0	1	1	0	0	右移
	1	1	1	0	1	0	右移
	2	1	0	1	0	0	右移
	3	0	1	0	0	0	右移
	4	1	0	0	1	0	右移
	5	0	0	1	1	1	置数
无效 状态	6	0	0	0	1	0	右移
	7	1	1	1	0	0	右移

循环的功能是通过在“001”状态置数“011”实现的。自启动功能是通过右移进入有效状态实现的。

2. 逻辑化简

(1) 74161

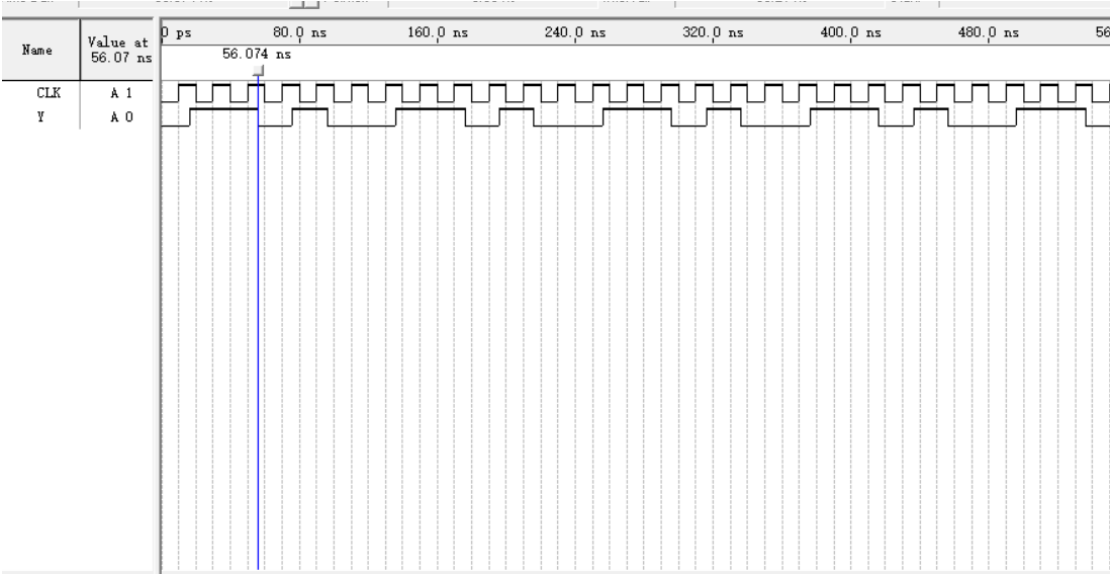
分析可知，该计数器是一个模 6 计数器。只需将“0”~“5”对应的二进制码分别接入对应输出即可。本实验采用同步置 0 的方式设计逻辑电路，即在计数器输出“5”的后一个 CLK 上升沿将预置数“0”同步置进计数器中。逻辑函数式参考模 4 计数器，输出的转化同样用 74138 实现。

(2) 74194

分析可知，初态为 011（前两位为“01”的原因是：有效状态中已经含有 010，故顺序

可以看到，初状态为 Y=0；延迟约 6.729ns；Y 正确显示了时序波形（101001）。

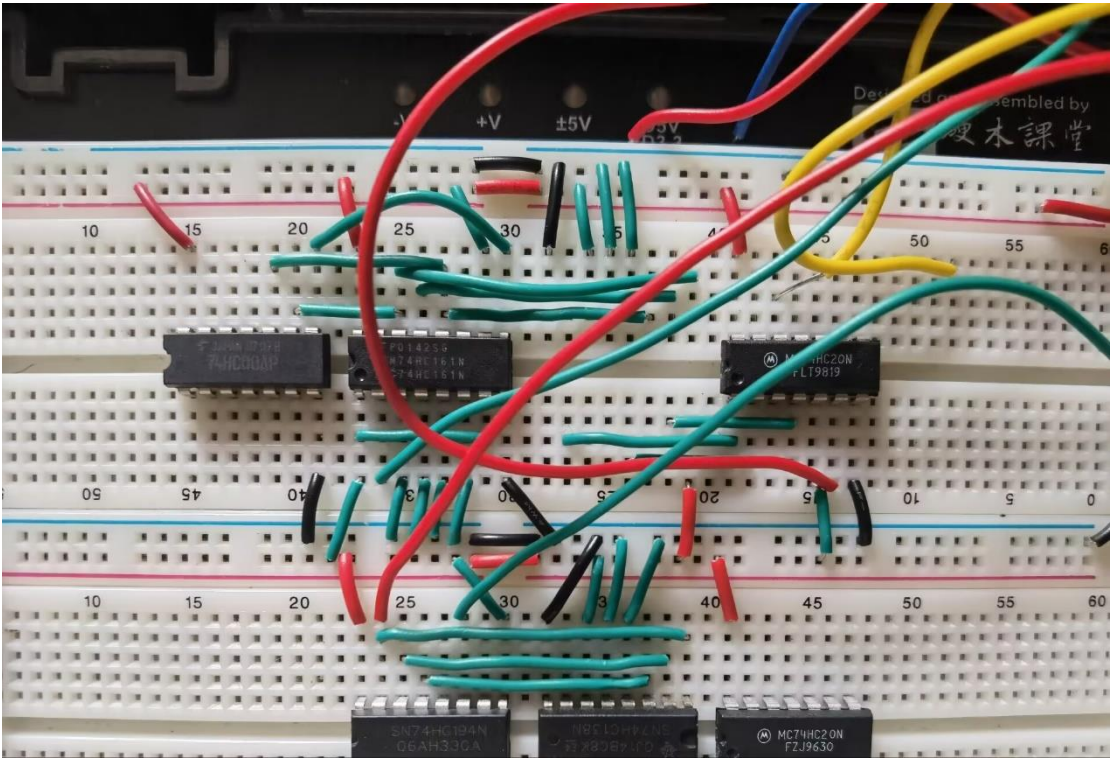
74194 实现:



可以看到，初状态为 Y=0，随后置数 Y=1，然后正确显示了波形；延迟约 6.074ns；Y 正确显示了时序波形（101001）。

5. 硬件连接图

74161:



（空缺位置为上下颠倒的 74138 芯片，红线接 16 脚，黑线接 8 脚，可以确定其位置。由于编辑报告时电路已拆卸，因此仅保留了这张图）

74194:

3、 4 位并行输入-串行输出曼切斯特编码电路（第 12 周课内验收，基础部分占 70%，扩展部分占 30%）

在电信与数据存储中，曼彻斯特编码（Manchester coding），又称自同步码、相位编码（phase encoding, PE），它能够用信号的变化来保持发送设备和接收设备之间的同步，在以太网中，被物理层使用来编码一个同步位流的时钟和数据。曼彻斯特编码用电压的变化来分辨 0 和 1，从高电平到低电平的跳变代表 0，而从低电平到高电平的跳变代表 1。信号的保持不会超过一个比特位的时间间隔。即使是 0 或 1 的序列，信号也将在每个时间间隔的中间发生跳变。这种跳变将允许接收设备的时钟与发送设备的时钟保持一致，图 3.2 为曼切斯特编码的例子。

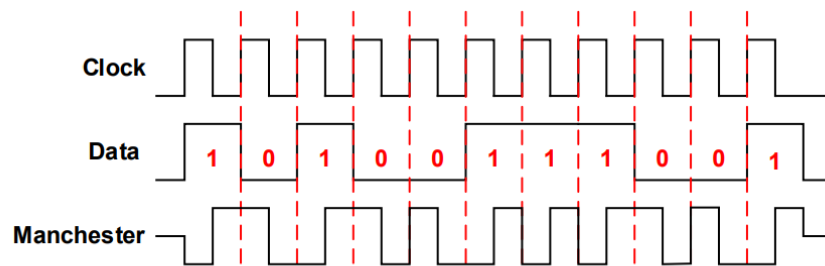


图 3.2 曼切斯特编码

设计一个电路，它能自动加载 4 位并行数据，并将这 4 位数据逐个串行输出（高位在前），每个串行输出位都被编码成曼切斯特码，当 4 位数据全部传输完成后，重新加载新数据，继续传输，如图 3.3 所示。

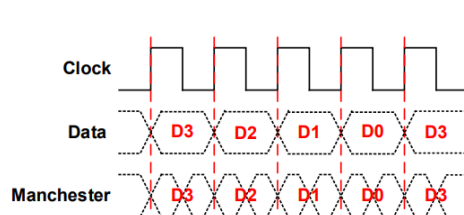
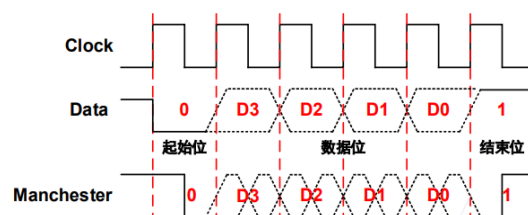


图 3.3 基础部分波形



3.4 扩展功能波形

- (1) 写出设计过程，画出电路逻辑图，设计时要注意不允许手动加载数据。
- (2) 加入 TTL 连续脉冲，用示波器观察观察并记录时钟脉冲 CLK、串行数据输出端的波形。
- (3) 给串行数据增加起始位和结束位，其中起始位为“0”，结束位为“1”，起始和结束位同样要编码成曼切斯特码，波形图参看图 3.4（扩展部分，选作）

0 输入信号与输出信号编码

输入信号

曼切斯特码：用 CLK 作为时钟信号。

输出信号

曼切斯特码：用 M 作为输出信号，Y 作为曼切斯特码。高电平代表信号“1”，低电平代表信号“0”。

1. 设计思路及真值表

用 74194 与 74161+74138 组合的形式实现该电路。其中，194 负责加载数据。161+138 负责计数 0~5，计数 1~4 时串行输出数据，计数为 0 时为起始位，计数为 5 时为结束位。

置 0 或置 1 的思路是通过逻辑“0”在与运算中的优先级实现的：逻辑 0 与任何数据的与运算后都会得到逻辑 0，即可实现起始位强制置 0 的功能。同理，要实现逻辑 1，只需将数据与逻辑 0 作与非即可。

74161+74138 实现模 6 计数器。

具体实现如下：

计数模块：

计数模块即为一个简单的模 6 计数器。将 161 的低 3 位接到 138 的对应 3 位二进制输入，即可实现 161 计数的二进制输出与 138 十进制输出的转换。将 138 的 Y_5' 通过反相器接回 161 的 LD 脚，使得计数为 5 时同步置 0，即可实现模 6 计数的功能。

产生数据模块：

采用 74194 实现产生数据的功能模块。设并行数据输入为 dcba，从 Q_D 输出串行数据，方便起见，用 161 计数的序号标注 QDQCQBQA 的组合状态顺序（即状态 1~4 为数据输出状态，状态 0 为起始位，状态 5 为结束位）。输出数据时，只需在 CLK 高电平到来时右移即可。

4 位数据输出完成后，下一个 CLK 到来时，194 进入状态 5。通过置数功能实现结束位的数据处理。即：结束位时，194 的数据状态应是 dcba。

下一个 CLK 到来时，194 进入状态 0。通过保持功能实现起始位的数据处理。即：起始位时，194 的数据状态将保持结束位的状态，仍应是 dcba。

根据上述分析所得状态图如下：

表 4 74194 状态图

	顺序	Q_D	Q_C	Q_B	Q_A	SR/SL	S_0	S_1	功能
有效 状态	1	d	c	b	a	φ	1	0	右移
	2	c	b	a	0				
	3	b	a	0	0				
	4	a	0	0	0				
结束位	5	0	0	0	0		1	1	置数
起始位	0	1	1	1	1		0	0	保持
无效 状态	6	φ					1	0	右移
	...								
	15								

其中 φ 代表无关量。

现将结束位、起始位与计数模块关联。由设计思路，结束位时应由 Y_5 控制 $S_0 S_1$ 输入组合为 (1, 1)；起始位时应由 Y_0 控制其输入组合为 (0, 0)；输出数据时 $S_0 S_1$ 的组合应为 (1, 0)。因此， S_0 应接 Y_0 ， S_1 应接 Y_5' 。这里 $Y_5 Y_0$ 均指引脚。

输出模块：

由上文可知，可以通过与逻辑 0 作与运算实现起始位与结束位。

161+138 计数 0000 (0) 时判断为起始位，138 的 Y_0 脚输出为 0，将其与 Q_D 作与运算，即可实现起始位置 0；计数 0101 (5) 时判断为结束位，138 的 Y_5 脚输出为 0，将其与 Q_D 作与非运算，即可实现结束位置 1；计数为 1~4 时，由于此时 Y_0 及 Y_5 脚均输出 1，与 Q_D 作与或非运算时不会影响输出。

因此，最终输出 M 应是 Q_D 输出 Q 与 138 Y_0 脚的与，及与 Y_5 脚的与非。与和与非无先后顺序要求。

同时，考虑到过了一次与非门，为保证输出 M 仍是原数据而非反数据，应给 Q_D 脚添加一个反相器。这就是最终输出 M 的处理思路。

2. 逻辑化简

根据上述分析可列出部分逻辑关系。由于 161+138 的逻辑部分较为常见，在此只列出最终输出 Y 与 74194 芯片上两个 S 脚的逻辑函数。

记 Q_D 输出为 Q_D ，则最终输出 M 的逻辑函数如下：

$$M = \overline{\overline{Q_D} \cdot \overline{Y_5}} \cdot Y_0$$

S_1 与 S_2 逻辑函数如下：

$$\begin{cases} S_0 = Y_0 \\ S_1 = \overline{Y_5} \end{cases}$$

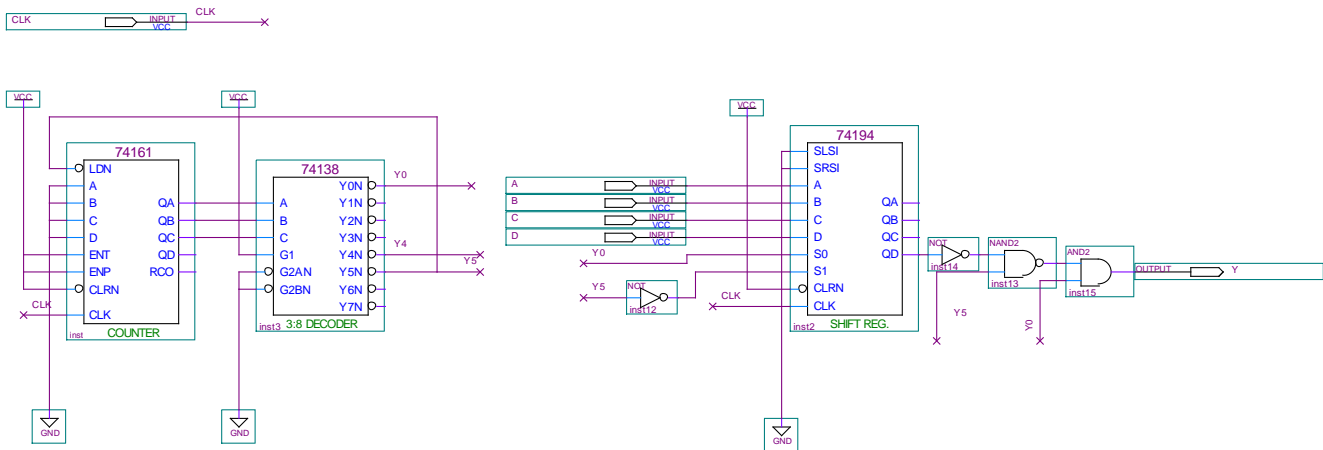
其余部分按逻辑功能正常接入即可。

将最终输出 M 编码为曼切斯特码：

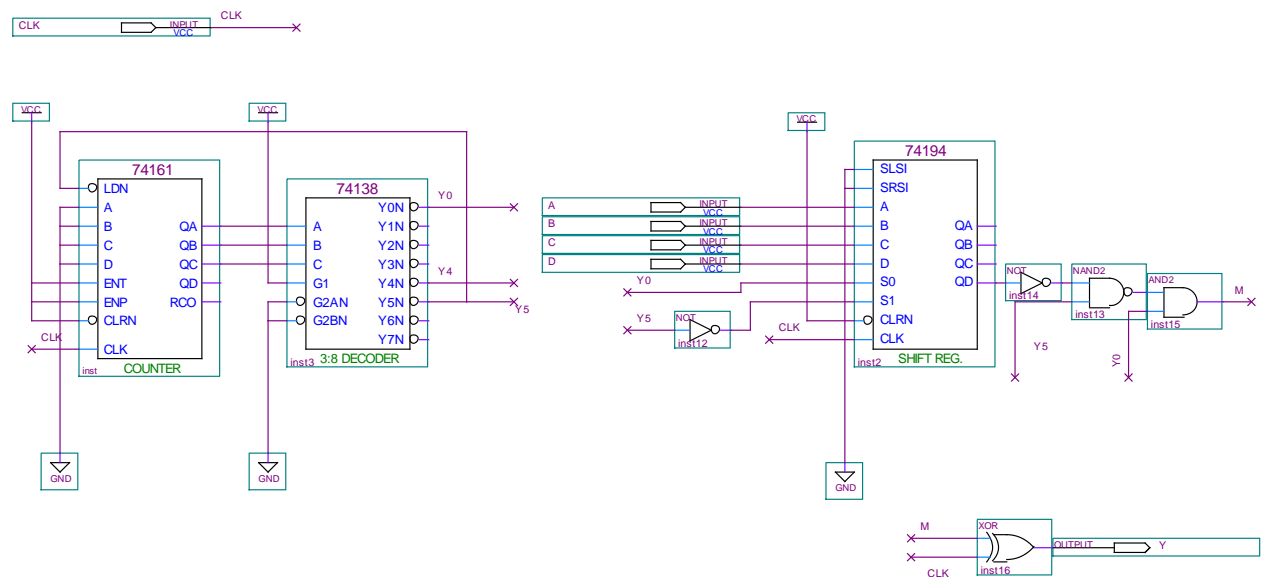
$$Y = M \oplus CLK$$

3. 逻辑电路图

不考虑曼切斯特码，仅只考虑并行输入、串行输出、开始位 0 与结束位 1 的电路图如下：

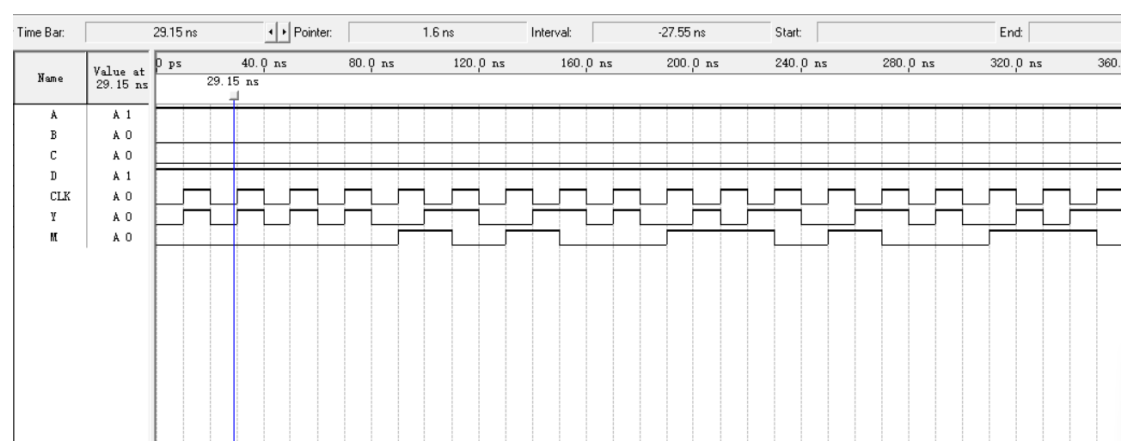


将该电路输出编成曼切斯特码，即将原输出与 CLK 信号异或。由于起始位与结束位也要编成曼切斯特码，故可以直接异或，不需要考虑 161 计数为 $Y_1 \sim Y_4$ 时才异或。如下：



4. 仿真

功能仿真：

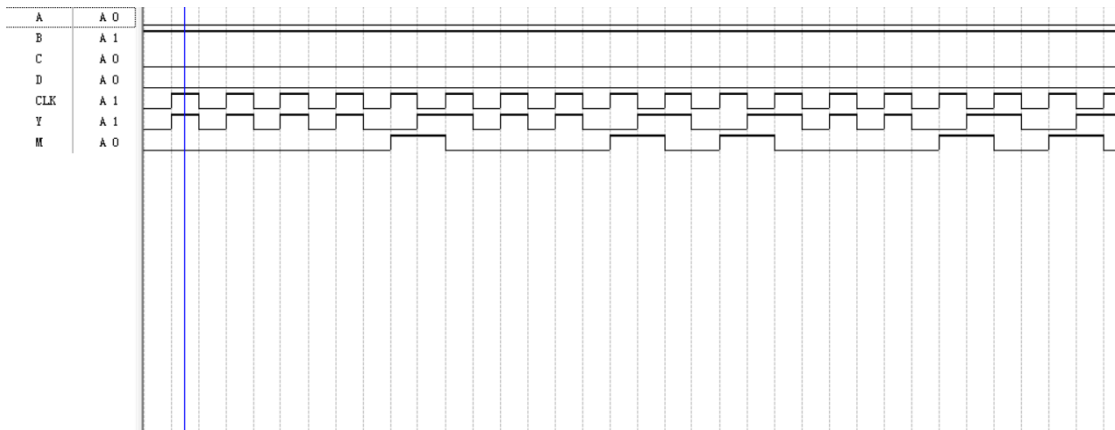


这里输入数据设定为了 1001。可以看到，数据输出 M 在第一个序列周期内无数据输出，是因为初状态时没有对 74194 置数，默认数据为 0000。第二个周期起，首先输出 0 信号，接着输出 1001 数据，最后输出 1 信号，标至第二个周期结束。第三个周期重复了第二个周期的输出。

同时，曼切斯特码 Y 正确对 M 与 CLK 作了与非。取 t=0ns 为例，M=0，CLK=0 时 Y 输出为 0，CLK=1 时 Y 输出为 1，即 CLK 与 M 同图像。

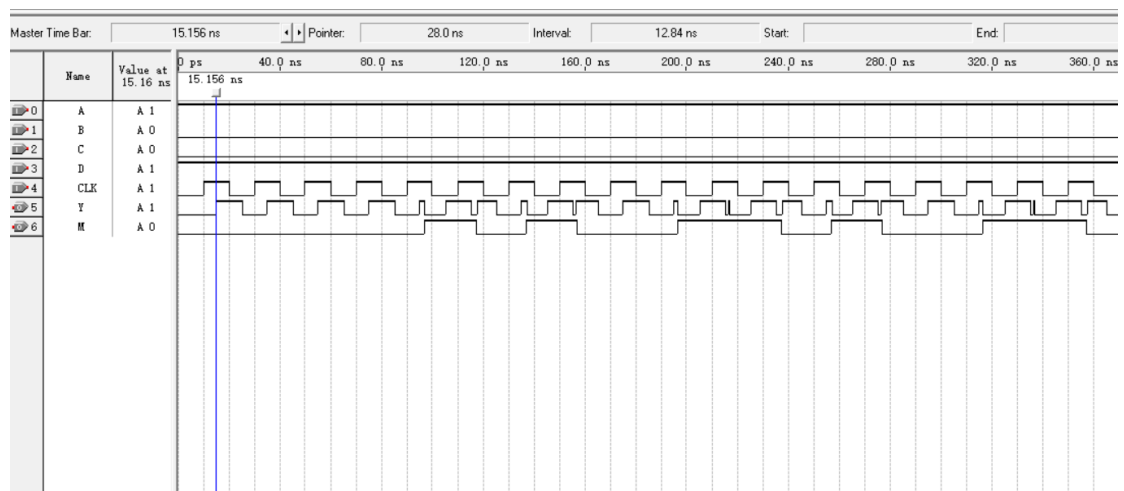
因此，该电路图功能正确。

更改数据为（0010）：



同样，M 正确输出了 0-0010-1 的序列，Y 正确进行了与非。

时序仿真（1001 为例）：

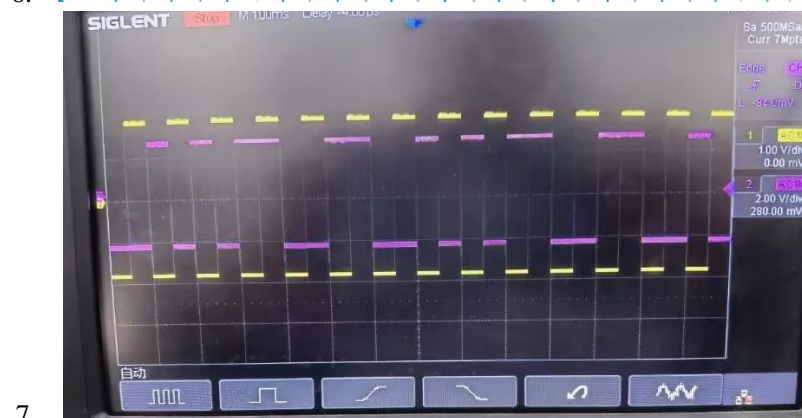
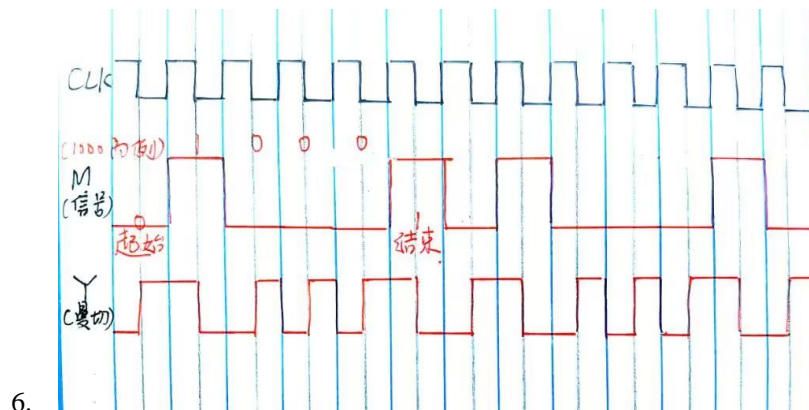


可以看到，曼切斯特码输出延迟约为 5.156ns。观察波形可以发现，数据输出 M 波形正确，为 0-1001-0 循环；曼切斯特码 Y 输出正确，实现了数据与 CLK 的同步；且二者均无毛刺。

5. 实验记录

实验步骤：首先利用脉冲按钮进行静态验证。按下按钮后，计数器输出从 0~5 循环，信号 M 依次输出起始位 0、信号序列、输出位 1，每次按下按钮信号改变一次信号，计数器计数+1，信号序列输出正确。通过改变置数验证缓存功能：设定信号为 0000，在计数器为 0 时（起始位）改变信号为 1111，输出信号仍为 0000，能够缓存。曼切斯特码的结果正确。静态验证结果符合功能。

再利用双踪示波器进行动态验证。利用信号源产生频率 1kHz、高电平 5V、低电平 0V 的方波作为 CLK 信号，显示(CLK,Y)、(CLK,M)波形，记录结果。这里的信号是以“1000”为例的。



波形说明，连续 CLK 信号输入时，M 循环输出“010001”信号，Y 循环输出对应的曼切斯特码。动态验证结果符合功能。

经检验，该电路**能够自启动**，从非 0~5 状态开始计数时能成功进入有效状态。
 遇到的问题及解决方法：无。

三、实验仪器

FPGA 板

四、实验小结

本次实验均较为成功。学习到了如何进行动态验证与静态验证，并且在实验四中（序列检测器）已经能够自行设计静态验收方案，与老师上课要求的验收方案几乎一致，算是一个小进步。

进行了本次实验后，已经能熟练设计有条件循环的计数器了，可以很轻松地得到对应功能需要的计数器，并且对寄存器的缓存功能有了更深的了解。例如，下一个实验中（序列检测器的实验我在曼切斯特码验收前就已经完成了分模块的功能调试），在试图拓展功能、整合静态与动态“1011011011100100”验证功能时，发现输入的序列信号与 CLK 信号不同步，但会计入最终输出。于是，想到了 D 锁存器的特性，将序列输入经过一个 D 锁存器再输出，实现了 CLK 信号与输入信号的统一。

综合来说，本次实验我各方面均有了更深刻的理解。

五、参考资料

《数字逻辑设计实践 2023 年教学计划 A_V1.0》